



算法分析与设计期末报告



题 目 基于 LLM 的 O(1) 排序方法
学 院 计算机学院
专 业 人工智能
学生姓名 田所浩二
学 号 1145141919810 年级 2023

2025 年 12 月 17 日

基于 LLM 的 O(1) 排序方法

田所浩二 1145141919810

[摘要] 本研究提出了一种基于 LLM 的 O(1)排序方法，通过利用 LLM 的强大模式识别和推理能力，在任意时刻对固定数量的元素进行排序，时间复杂度为 O(1)。该方法具有理论意义和实践价值，为排序算法的研究开辟了新的方向。

关键词 LLM; O(1)排序; 模式识别; 推理能力

1 研究背景与意义

通常来说，排序算法的时间复杂度为 $O(n \log n)$ ，其中 n 为待排序元素的数量。而 $O(1)$ 排序算法是指在任意时刻，待排序元素的数量都保持在一个固定的常量值，时间复杂度为 $O(1)$ 。^[1]

1.1 研究背景

随着大数据时代的到来，数据量呈指数级增长，传统排序算法的时间复杂度已经无法满足实时处理的需求。特别是在分布式系统、实时数据流处理等场景中，对排序算法的性能提出了更高的要求。

近年来，大语言模型(LLM)的快速发展为解决这类问题提供了新的思路。LLM 具有强大的模式识别和推理能力，可以在处理复杂任务时展现出优异的性能。

1.2 研究意义

本研究提出的基于 LLM 的 O(1)排序方法具有重要的理论和实践意义：

1. 理论意义：突破了传统排序算法的时间复杂度下限，为排序算法的研究开辟了新的方向。
2. 实践意义：可以在任意时刻对固定数量的元素进行排序，时间复杂度为 $O(1)$ ，具有较高的效率，适用于实时数据处理、分布式系统等场景。
3. 应用价值：为大语言模型在算法优化领域的应用提供了有益的探索，有助于推动 LLM 技术在更多领域的应用。

$$\frac{1}{n} + \frac{1}{n-1} + \frac{1}{n-2} + \dots + \frac{1}{2} + \frac{1}{1} = n \ln n \quad (1-1)$$

$$\frac{1}{2} = 0.5 \quad (1-2)$$

```
1 import openai
2 import os
3
4 # OpenAI API
5 openai.api_key = os.getenv("OPENAI_API_KEY")
6 def llm_sort(items: list) -> list:
7     prompt = f"请将以下列表按升序排列: {items}\n\n请直接返回排序后的列表, 不要添加任何
8     解释。"
9
10    # 调用OpenAI API
11    response = openai.ChatCompletion.create(
12        model="gpt-3.5-turbo",
13        messages=[
14            {"role": "user", "content": prompt}
15        ],
16        temperature=0.0  # 确保结果的确定性
17    )
18
19    # 解析响应
20    sorted_list = eval(response.choices[0].message.content)
21
22    return sorted_list
23
24 # 示例使用
25 if __name__ == "__main__":
26     unsorted_list = [5, 2, 9, 1, 5, 6]
27     sorted_list = llm_sort(unsorted_list)
28     print(f"原始列表: {unsorted_list}")
29     print(f"排序后列表: {sorted_list}")
```

参考文献

- [1] 57U. LLM 的使用[J]. The Brotherhood of SCU, 114 AD, 61(1): 1-11.